Hindcast Robot
User's Manual

Version: OHD-CORE-CHPS-4.4.a
Release Date: 16 May 2016

National Weather Service
Office of Hydrologic Development

Table of Contents

1 Overview

The Hindcast Robot is a tool that allows for automating interactions with hindcasting SAs necessary in order to generate hindcasts.

This document provides instructions for setting up and executing the Hindcast Robot.

Using the robot to generate the hindcasts has some dramatic advantages. First, running hindcasts makes use of fixed, independent procedures which can be time consuming to perform and may need to be performed repeatedly. For example, the hindcast period may need to be chopped into several short periods for the following reasons:

- To generate hindcasts simultaneously with multiple SAs and computers to fully utilize the available CPUs.
- To avoid problems with available disk space and files becoming so large that performance is impacted. Specifically, the localDataStore, log.txt, and directory containing exported files can increase in size significantly when generating hindcasts.
- To generate CFSv2 reforecasts, hindcast periods must be defined to avoid leap days (see the *HEFS Hindcasting Guide* for more information). For example, to generate a hindcast that includes the CFSv2 forecast source for MEFP for a period of 1985 to 1988, the overall hindcast period can be broken down as follows:

  01-01-1985 – 02-28-1988
  03-02-1988 – 02-28-1992
  03-02-1992 – 02-28-1996
  03-02-1996 – 12-31-1998

The need to perform hindcasting for many small periods can dramatically increase the work load for the user. Furthermore, it increases the chance of mistakes such as typos. In contrast, using the Hindcast Robot reduces the burden. For each run, an input control file specifying the starting hindcast T0, ending hindcast T0, and other information is created by the user. Then the user can execute the Hindcast Robot with one or more control files. The robot can be executed in two modes:

- *parallel mode*: All provided control files are processed (nearly) at the same time using multiple hindcasting SAs.
- *series mode*: All provided control files are processed to completion, one after another, in order.

Other advantages of using the Hindcast Robot are as follows:

- During hindcasting due to many factors, such as too much load on the machine, timeout issues and others, the hindcasting SA can sometimes skip hindcast T0s. At the end of a run, the Hindcast Robot checks for any missing days and re-runs them.
- The Hindcast Robot can detect errors:

- If a workflow fails, the robot will detect it and close FEWS. The robot then moves on to the next job if there is another control file waiting in the queue.
- If a workflow execution is stalled for more than a set period of time, the Hindcast Robot will detect it and close FEWS.

1.1 *Notation*

Within this document, the following notation is used:

- All command line entries, control options, and environment variables are in this font.
- All important terms defined in the Section 1.2, Terminology, are *italicized*.

1.2 *Terminology*

- *hindcasting standalone (SA)*: A standalone used to generate hindcasts for HEFS components, setup in Section 1.4.
- *hindcast workflow*: The workflow to be executed when generating the HEFS hindcast, after the warm states have been created.
- *hindcast T0*: A forecast time (T0) for which a hindcast is to be generated.

1.3 *Directories of Note*

The following directories will be referred to in the instructions provided below:

- <*region_dir*>: A *hindcasting standalone* (see Section 1.4) region home directory.
- <*export_root_dir*>: The directory under which PI-timeseries files will be exported for verification by EVS; typically <*region_dir*>/Export.
- <*tar_root_dir*>: The directory to where the CHPS release has been unpackaged.
- <*robot_dir*>: Where the contents of <*tar_root_dir*>/hindcastRobot will be moved as part of Section 2.1.

1.4 *Pre-installation Steps*

1. Create all *hindcasting SA*s to be used within the execution of the Hindcast Robot. See the *Hindcasting Guide* for more information.

1.5 *Release Package*

As part of installing the CHPS release, the release package was acquired and untarred in a directory referred to as <*tar_root_dir*>. Within this document, only the contents of the subdirectory hindcastRobot are used. The hindcastRobot subdirectory contents are as follows:

<*tar_root_dir*>/hindcastRobot/…
- fews_hindcast_robot.jar
- sikuliLibs/…
  - gcc_4.7.3/
  - leptonica_1.6.9/
  - opencv_2.4.1/
  - tesseract-ocr_3.0.2/
- wmctrl

2 Setup

This section provides instructions for setting up the Hindcast Robot for execution.

The hindcast robot program is written in Java. It makes a call to the FEWS PI-Service to execute the *hindcast workflow*. Compared to the usual mechanism, i.e., execution through the **Manual Forecast Dialog** of the *hindcasting SA*, using the FEWS PI-Service is about 15% faster (based on experimentation using CHPS 5.1.1).

In order to achieve completely hands free execution, the Hindcast Robot uses a free MIT developed Java program called Sikuli, which makes use of open source software (tesseract-OCR, openCV and others). Library files (*.so files) for that software are provided in the sikuliLibs/ directory(see below). Environment variable LD_LIBRARY_PATH needs to be set so that these library files can be dynamically linked, instructions are provided in 2.3.

The Hindcast Robot requires a 32-bit Java version 1.7 or higher. The java version delivered for use with CHPS should be sufficient:

/awips/chps_share/jre/bin/java

2.1 *Copy the Release Directory to a Central Location (Required)*

Action: Identify a directory to permanently contain the contents of the directory <*tar_root_dir*>/hindcastRobot. It will be denoted <*robot_dir*> below. The directory should be a shared directory, visible to any machine that will be used to execute the Hindcast Robot.

Action: Copy the release directory into <*robot_dir*>:

```
cd <tar_root_dir>/hindcastRobot
cp –r * <robot_dir>/.
```

2.2 *Set the LD_LIBRARY_PATH (Required)*

Action: Execute the following commands:

```
export LD_LIBRARY_PATH=<robot_dir>/sikuliLibs/gcc_4.7.3:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=<robot_dir>/sikuliLibs/leptonica_1.6.9:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=<robot_dir>/sikuliLibs/opencv_2.4.1:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=<robot_dir>/sikuliLibs/tesseractocr_3.0.2:$LD_LIBRARY_PATH
```

Description: The LD_LIBRARY_PATH environment variable is used by the Sikuli software in order to find shared libraries (*.so files) that must be linked to the software at run time.

2.3 *Set the User's PATH (Required)*

Action: Set the PATH environment variable so that the executable command wmctrl can be found:

export PATH=<*robot_dir*>:$PATH

3 Executing the Hindcast Robot

This section provides instructions for the following:

- **Configuring an input control file.**
- **Executing the Hindcast Robot.**
- **Running the NXClient Virtual Desktop.**

3.1 *Configuring the Input Control File (Required)*

Action: Prepare one or more input control files for executing the Hindcast Robot.

One control file is needed for each *hindcasting SA* to be used. When the entire hindcasting period is divided into subperiods, one control file is needed for each subperiod. If you need to generate the hindcasts simultaneously (presumably in parallel using different machines), multiple *hindcasting SA*s need to be created.

Description: To execute the Hindcast Robot, at least one input control file must be prepared. Each input control file specifies which *hindcasting SA* to use, which FEWS script to use in order to start the SA, the start and end *hindcast T0*, etc. The file is a standard property file, where each line is of the format:

<property name> = <property value>

The required properties are as follows (In the 'Reqd?' column, a 'Yes" indicates if the property is required; a 'No' indicated it is optional). An example is provided below the table.

Property Name	Reqd?	Value Type	Description
SA_FOLDER	Yes	Text	Provides the <*region_dir*> of the *hindcasting SA* for the Robot to run.
FEWS_SCRIPT	Yes	Text	Defines the FEWS start script (e.g., fews_ohdPlugins.sh.rboff) to use in order to start the *hindcasting SA*. It is recommended to start the SA with the *rboff version of the start script.
WORKFLOW_ID	Yes	Text	The workflowId XML element defined for the *hindcast workflow* in the WorkflowDescriptors.xml configuration file.
FORECAST_LENGTH	Yes	Positive Integer	The length (# of days) of the hindcasts to generate. The value must be in the range between the minForecastLength and maxForecastLength associated with the *hindcast workflow* descriptor in the WorkflowDescriptors.xml configuration file.
START_T0	Yes	See Description	The date of the first hindcast T0; format: MM-DD-YYYY HH:00:00

END_T0	Yes	See Description	The date of the last hindcast T0; format: MM-DD-YYYY HH:00:00
EXPORT_FOLDER	No	Text	**The default value is <*region_dir*>/Export** Defines the parent directory in which all the files exported during hindcast are placed. All exported files must be placed within one of its subdirectories, but cannot be placed any lower in the directory tree. For example, if all the result files are inside the three directories: <*region_dir*>/Export/hefs <*region_dir*>/Export/mefp_precip <*region_dir*>/Export/mefp_temp Then set the value to: <*region_dir*>/Export/ After a hindcast run is finished, the Hindcast Robot counts the total number of export files within all the sub-directories for each T0 to see if any random skipping has occurred. If every *hindcast T0* has the same number of result files, the run is finished; if there are one or two *hindcast T0*s without the correct number of result files, the robot will automatically restart FEWS and run only those days.
DAY_INTERVAL	No	Positive Integer	**The default value is 1.** The interval between consecutive *hindcast T0*s.
SCRIPT1	No	Text	If provided, the shell script specified as the property value will be executed when the hindcast run is finished. This allows users to insert customized tasks into the Hindcast Robot program. For example, a script could be written to send an email to the user to notify them when the run has finished. As another example, a script could be written compress the result files to minimize the disk space used. The Hindcast Robot will wait for this script to complete before moving on.
SCRIPT2	No	Text	As SCRIPT1, but the Hindcast Robot will not wait for this script to complete; it will run it in the background and the robot will continue with its processing immediately.
DEBUG	No	0 or 1	**The default value is 0.** When specified with a value of 1, debug messages will be printed out as the Hindcast Robot executes. If 0, then no debug messages are output.

EXAMPLE: Input Control File

```
SA_FOLDER=/home/joe/hindcast/cnrfc_sa_job1
FEWS_SCRIPT=/home/joe/hindcast/bin/fews.sh.rboff
WORKFLOW_ID=HEFS_EndToEnd_Forecast
FORECAST_LENGTH=15
START_T0=01-01-1987 12:00:00
END_T0=12-31-1987 12:00:00
EXPORT_FOLDER=/home/joe/hindcast/cnrfc_sa_job1/Export/ (Optional, default is Export/ under the SA)
DAY_INTERVAL=5
# SCRIPT1=/home/joe/exampleScript1
# SCRIPT2=/home/joe/exampleScript2
DEBUG=1
```

3.2 *Run the Hindcast Robot to Generate Hindcasts (Required)*

Action: Execute the following command to run the Hindcast Robot:

/awips/chps_share/jre/bin/java -jar <*robot_dir*>/fews_hindcast_robot.jar <input control file 1> [<input control file 2> ...] [parallel] [scan]

Note the following:

- The Java command used above refers to that delivered with CHPS. If another Java command is to be used, the change the command line above appropriately.
- <input control file #> are the input control files to be processed (one or more). Each control file is processed to completion, including reruns, unless the parallel command line argument is included.
- If parallel is included as the last or second to last argument (if scan is included) of the command line, then, after the hindcasting job indicated by a control file is started, the Hindcast Robot will start the next job in the sequence without waiting for the previous job to complete.
- If scan is included as the last argument, then the Hindcast Robot will not generate any hindcasts. Instead, it will examine the directory indicated by the EXPORT_FOLDER control file option and report for which *hindcast T0*s missing files are found.
- Execute the command above without any arguments after fews_hindcast_robot.jar to see help information.

When the robot launches a *hindcasting SA*, do not use the mouse and the keyboard until the robot has started the workflow and minimized FEWS.

Description: As the Hindcast Robot processes each input control file, it does the following:

1. If the port number has been specified in the hindcasting SA's sa_global.properties file by the line "PiServicePort=####", the Hindcast Robot will comment out that line beforehand and let FEWS use whichever port number is available, starting from 8100.
2. Launch FEWS against the specified *hindcasting SA* using the specified FEWS start script as defined in the input control file.
3. If there is a yellow triangle at the FEWS toolbar at the bottom indicating that error(s) have occurred in the past, then click it, press the F12 key, press key 'I' to acknowledge it so that it changes to a green ball; if it is already a green ball, then go to step 3 directly.
4. Retrieve the FEWS PI-Service port number from the log.txt log file.
5. Call the FEWS PI-Service using that port number on the localHost and execute the *hindcast workflow* with the start time and end time specified in the control file.
6. After 30 seconds, minimize the FEWS interface as a shade.
7. After the run has finished, close its corresponding FEWS session (even when there are several FEWS open at this time) and check if all the exported file in the directory specified in

the control file have been created for each day from starting *hindcast T0* to ending T0. If any day's result files are missing, the robot will print it out on the terminal and will re-start the *hindcasting SA* in order to rerun the hindcast for that specific day. However, if there are more than 10 days found to be missing files, the Hindcast Robot assumes that a problem has occurred; it will note it and quit.

- A screen saver will block the Robot from starting FEWS and performing other actions, so turn off the screen saver before starting a Hindcast Robot. Alternatively, you may use virtual desktop, like NXClient explained next, which we strongly think is the best way to run hindcast.
- Each time the Hindcast Robot is run, a temporary directory called "libs" will be created by Sikuli in <*robot_dir*>. If the user does not have write permissions for the directory, the Hindcast Robot will fail.
- When multiple Hindcast Robots are executed, each needs to use the computer keyboard and mouse to enter information in the *hindcasting SA* FEWS interface. When one robot has seized control of the computer, it creates a temporary lock file called robot_desktop_lock_file in the user's home directory, causing other jobs (if any) to wait. When the hindcasting has been started, the robot minimizes the interface for the *hindcasting SA* as a shade and deletes the file robot_desktop_lock_file, allowing other jobs to commence.
- When executing the Hindcast Robot on a single machine but using multiple terminals (i.e., so that each robot can be run from a different machine fully utilizing computer resources), it is possible that while still typing the commands in some of the terminals, the Hindcast Robot execution from other terminals has already started its FEWS *hindcasting SA* session and is using the keyboard and mouse. This makes it impossible for you to finish typing the rest of the commands on other terminals windows. To deal with this problem, first manually create the lock file robot_desktop_lock_file in the user's home directory by using touch command, forcing all the robots to wait:

```
cd ~
touch robot_desktop_lock_file
```

After finishing typing all your commands on the terminals, manually delete the lock file allowing the different Hindcast Robots to start.

3.3 *Running in NXClient Virtual Desktop (Optional)*

Action: Use the NXClient Virtual Desktop to avoid issues due to screen savers or user interactions with the interface conflicting with those of the Hindcast Robot.

Description: The NXClient Virtual Desktop is available on AWIPS machines. For a single user, only one session can be started on one machine. Inside the single NXClient session, you can run all required *hindcasting SA* sessions.

Running inside a virtual desktop has several advantages:

- Once all of the commands have been entered in terminals within the virtual desktop, the virtual desktop can be closed and the user can log-off of the machine. Running in the background in this manner, the Hindcast Robot sessions will continue performing their activities as usual. Unless the computer shuts down (or reboots), the robots and *hindcasting SA*s run will not stop. In contrast, if you run the hindcast on a regular desktop and if the run needs several days to finish, there may be other interferences that cause the robot or *hindcast SA*s to freeze or halt.
- You may log back to the virtual desktop from any computer to inspect the status of the executing Hindcast Robots.
- Virtual desktops avoid the screen saver issue described in the previous section.
- When running inside the virtual desktop, the Hindcast Robot will use the keyboard and mouse inside the virtual desktop, not on the user's machine. Thus, even while examining the progress of the virtual desktop, the user can use the keyboard and mouse on the regular desktop.